

Yazılım Proje Yönetimi:

- Proje yönetimi bilgi ve deneyimi
- Genel yönetim bilimi
- Yazılım mühendisliği teknikleri

Etkin Proje Yönetimi: 3P->People-Problem-Process

People-İnsan Kaynakları: Seçim, performans yönetimi, eğitim, kariyer geliştirme, organizasyon ve iş tasarımı, ekip kültürü geliştirme konuları üzerinde durulmalıdır.

Problem: Bir proje planlanmadan önce amacı ve alanı belirlenmeli, alternatif çözümler düşünülmeli, teknik ve yönetim kısıtları tanımlanmalıdır.

Process-Süreç: Yazılım geliştirme sürecinde gerçekleştirilen işlemler, esas işlemler ve bunların gerçekleşmesini destekleyen işlemlerden oluşur.

*Bu üç kurala **Product** eklenerek 4P kuralı da tanımlanabilir. Proje sonunda ortaya çıkacak ürünü de tanımlamak önemli!*

Yazılım mühendisliği, yazılım ürününü oluştururken, mühendislik yaklaşımı uygulamakla ilgili olan teknikler toplamını tanımlamak için kullanılan terimdir. Mühendislik yaklaşımı, yönetme-maliyet hesabı, planlama, modelleme, analiz etme, tasarlama, gerçekleştirme ve bakım anlamındadır. Tüm bu adımlarda ölçme vardır.

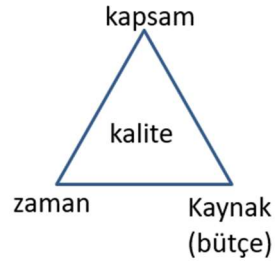
Yazılım geliştirme süreci:

Analiz / Tasarım / Geliştirme / Test

1.Kaliteli yazılım: donanımı iyi kullanan, performans, kullanıcı dostu, güvenlik, doğru çözüm ve bunları belirli bir hassaslıkla –reliable- yapan, reusable

2. Ekonomik: süre / kaynak planlaması / maliyet

3.Mühendislik araçları: tasarım / ölçme / hesaplama



Başarı göstergesi:

- Kapsam(problem)
- Zaman
- Kaynak

Başarısızlık nedenleri:

- Kapsam değişiklikleri
- İş aksaklığı

- Teknoloji ve yetkinlik eksikliği
- Yönetim desteği alınamaması

Yazılım Geliştirme Süreç Modelleri

-**Lineer Akış Waterfall (Şelale)** -> müşteriye geç yanıt alınıyor A->T->G->T

-**Prototip Model:** önce, gerçek hayatta kullanılmayacak bir model üstünde Analiz ve Tasarım adımları uygulanıyor, yalnızca müşteriden kolay bilgi alabilmek için yaratılmış bir prototip oluşturuluyor. [A->T]->G->T Prototipte kullanılan yapıların gerçek programda yansımaları/kullanılması gibi sorunları oluyor.

-**RAD (Rapid App. Dev. Model):** Önce bir sistem analizi yapıyoruz. Uygulama aynı anda yönetilebilecek farklı parçalara ayrılabilir kullanılır. [[Sistem Analizi]]->[İş modeli-veri modeli]->[Yazılım parçasının oluşturulması]->[Test] farklı ekipler bu modeli, her modüle uygulayabiliyor, sonra hepsi birleştiriliyor. Aynı anda çalışacak ekip bulmakta zorlanılabilir.

-**Spiral (Sarmal) Model:** Süreç adımları yavaş ama sağlam, kontroller çok fazla, maliyetli bir model. Önce kullanılmayacak bir prototip, sonra gerçek ürün, sonra bakım yöntemi belirleniyor. Prototip model ve waterfall

modelinin birleşimi gibi düşünülebilir.

-Incremental (Artımlı) Model ve Evolutionary (Evrimsel) Model

Incremental Model: Ürünün prototipteki gibi değil de, geliştirilecek projenin her adımda bir kısmının tamamlanması gerektiğinde kullanılır.

Evolutionary Model: Hem iyileştirme hem artım aynı anda, giderek daha fazla iyileştirme yapılır.

Agile (çevik) süreçler: Değişikliklere daha kolay uyum sağlayacak yapılar oluşturulur.

-Ekip içi iletişim

-Müşteri ile iletişim (müşterinin ekip üyelerinden biri olmasını sağlamak)

-Basitlik: çok karmaşık tasarımlar ile başlamamak

-Belgeleme: çok karmaşık değil, süreci anlatacak kadar yeterli

Agile modeller: XP (Extreme programming), Scrum, Kanban

XP: [Planlama]-[Tasarım]-[Kodlama]-[Test]->[Artımsal ürün] bu döngü tamamlandıktan sonra planlamadan teste kadar olan kısım tekrarlanır.

Planlama aşamasında, müşteriden öyküler alınıyor ve önceliklendiriliyor, büyük öyküler varsa kırılım yapması istenebiliyor. Tüm öykülerin 2 haftada bitirilebilecek şekilde planlanması isteniyor.

Tasarım aşamasında, basit tutmaya çalışıyoruz. CRC kartları oluşturuyoruz; Class-Responsibility-Collaboration yazılımı yaparken bir yandan da test yapılabilir (Pair Programming)

Scrum: Ekip içi ve müşteri ile iletişim ön planda. Öncelikle bir görev listesi oluşturuluyor. Koşullar planlanıyor. İşlevler atanıyor. Bu üç adım döngü halinde ilerliyor.

Yazılım Projesi belirli bir başlangıç ve bitiş noktası olan, amacı, kapsamı, bütçesi açıkça tanımlanmış ve bir defaya mahsus olarak gerçekleştirilen planlanmış aktiviteler bütünüdür.

Projenin tarafları/paydaşları

Müşteri: Faturayı ödeyen, kapsamı belirleyen

Sponsor

Proje yöneticisi: Zaman ve bütçe planlamalarını yapar, beklentileri yönetir

Kullanıcılar: son kullanıcılar

Teknik kişiler: işgücü, kaynak

Danışmanlar

Projenin aşamaları:

- 1-Projeyi anlama
- 2-Projeyi tanımlama
- 3-Projeyi planlama
- 4-Projeyi çalıştırma
- 5-Projenin kapatılması

1-Projeyi anlama: Niçin yapılıyor: *fayda-maliyet, fizibilite analizinin genel olarak yapılması, oluru var mı?*

Temeli nedir: *örn bir çalışanın oluşturduğu yeni bir fikir mi, regülasyon mu vs.*

Proje paydaşları kimler? (Yürütücü komite, müşteri py, sponsor, son kullanıcı, çalışacak teknik kişiler)

Proje destekleyecek birimler kimler?

Müşteri önceliği nedir?

2-Projeyi tanımlama: Teslim edilecekler.

Hedeflerin belirlenmesi

Kapsam tanımlama

Gözden geçirme ve Onay

Sistem geliştirme yaşam döngüsü: Waterfall, Iterative, Spiral vs.

3-Projeyi planlama:

Riskleri tanımlama ve yönetme: 1-Risk tanımlanır 2-Olma olasılığı nedir, etkisi ne olacak? 3-Önem derecesi 4-Önlem nedir? Belirli sürelerde bu 4 adım kontrol edilir ve bir çevrim içinde takip edilir. Risk adı-Tarih-Açıklama-Olasılık-Etki-Önem-Önlem başlıkları bulunan bir doküman ile takip edilebilir.

Varsayımların ve kısıtların listelenmesi: varsayımların doğru belirlenmesi proje planı için önemlidir. Hatalı varsayımlar projeyi kötü yönde etkiler.

Kalite yönetimi: kalite yönetiminin amacı yazılım projelerinde hataların azaltılmasıdır. Ölçülebilir olmalıdır. Gözden geçirme (review), takvim ve onay işlemleri nasıl olacak belirlenmeli. Üretilen yazılım kurumun gereksinimlerini karşılıyor mu, network güvenliği gereksinimleri karşılanıyor mu vs. Bu gözden geçirmeyi yapacak bir kurul varsa bu kurul kimlerden oluşacak?

Aktivitelerin tanımlanması ve listelerinin oluşturulması: İlerlemeyi izleyebilmek için proje küçük, yönetilebilir aktivitelere bölünmelidir. WBS oluşturulabilir.

Aktiviteler arası bağımlılıkların belirlenmesi: Finish-Start (en çok kullanılan, aktivitenin başlaması için önceki aktivitenin tamamlanmış olması gereklidir), Finish-Finish (aktivitenin bitmesi için önceki aktivitenin tamamlanmış olması gereklidir), Start-Start (aktivitenin başlayabilmesi için önceki aktivitenin başlamış olması gereklidir), S-F (çok kullanılmıyor)

Efor tahminlemesi: projenin tamamlanması için ne kadar efor gerekli tahmin edilir.

Zaman çizelgesi: Takvim hazırlama ve milestone'ların belirlenmesi yapılır. PERT (Program evaluation and review technique, aktivite süreleri belirsiz olduğunda) veya CPM (Critical Path Method, aktivite süreleri tanımlı ise) kullanılabilir. Çeşitli programlar yardımı ile Gannt şeması kullanılarak en kritik yolun belirlenir. Süre için, (iyimser + kötümser + en olası) / 6 = süre tahmini yapılabilir.

Proje kaynaklarının değerlendirilmesi ve atanması: Belirli aktivitelere belirli kişiler atanır. Kişilerin sürelerinin tamamı mı yoksa bir kısmı mı kullanılacak belirlenir. Toplam iş gücü gözden geçirilir, yeni kaynak tedariki yapılır.

Bütçe planlaması: proje başarısının en önemli 3 ölçütünden biridir (bazen en önemlisi), diğer ikisi takvim ve kapsam. Maliyet-fayda analizinin maliyet tarafı burada yapılır. Bir proje bütçesi dört tür maliyet kaleminden oluşur: personel ücretleri, giderler, sermaye maliyetleri ve genel ofis genel giderleri.

Dokümantasyonun tanımlanması

Uygulama aşamasının planlanması: eğitimler, devir-teslim nasıl olacak belirlenir.

Proje kapanışının planlanması: Kullanıcı kabul testleri sonrası signoff alınması için gerekli koşulların tanımlanması gereklidir.

Ekipler arası iletişimin planlanması: bilgi aktarımı nasıl olacak, iletişim hiyerarşisi nasıl olacak, performans raporlaması nasıl olacak, diğer kaynaklar veya kurumlarla iletişim planlaması

Proje planının yazılması: Tüm plan yapıldıktan sonra yazılı hale getirilir.

4-Projeyi çalıştırma (running): izleme ve kontrol. Projenin gerçekleştirilmesi ve takibi aşaması.

Ekibin oluşturulması ve takibi: Ekip, ortak bir hedef için çalışan bir grup insandır. Proje yöneticisinin görevi, çalışmaların düzgün ilerlediğinden ve hedef dışına çıkmadığından emin olmaktır. Ekipler-kişiler arası iletişimdeki sorunlar giderilmelidir.

Aktivite ilerlemesinin izlenmesi: Plandaki süre-kaynak-iş kapsamı ile gerçekleşenlerin sürecin karşılaştırılması yapılır.

Risklerin yönetimi: Proje planında belirlenen riskler belirli periyotlarda gözden geçirilir, örn. haftalık. Yok edilen risklerin tekrar ortaya çıkmadığından, kabul edilen risklerin derecesinin değişmediğinden emin olunur.

Kapsam değişikliklerinin yönetimi: Kapsam değişiklikleri bütçeyi ve takvimi negatif etkiler. Proje tanımlama aşamasında belirlenmeyen hedefler değişiklik yönetimi ile takip edilir. Proje kickoff toplantısında proje kapsamının iyi anlaşıldığından emin olunmalıdır.

Kalitenin izlenmesi ve yönetimi: Proje planında tanımlanan kalite yönetimi adımlarının gerçekleştirildiğinden emin olunur. Gözden geçirme adımlarının tamamlanması ve bunların çıktılarının kabul edilmesi sağlanmalıdır.

Alt yöneticilerin yönetimi: Alt yöneticiler genelde taseronlar veya tedarikçilerdir. Bu paydaşlarla yapılan sözleşmelerin yönetimi de bu adımda yapılır. Yazılım, donanım, eleman kaynağı da bu kısma dahil edilebilir.

Ekip ve Müşteri toplantıları izleme: Ekip arasındaki iş birliğinin gelişmesi için önemlidir. Sorun çözme amaçlı değil, iletişimi kuvvetlendirme amaçlı yapılmalıdır. Ekip ve Müşteri tarafında yapılacak toplantılar belirli bir rutin izlemelidir.

5-Projenin kapatılması: Bir projeyi kapatmak, üç ana adımı içerir: projenin bittiğine dair müşteri mutabakatının sağlanması, proje sırasında öğrenilen derslerin alınması ve idari kapanışın tamamlanması.

Müşteri mutabakatının alınması: Planlama aşamasında tanımlanan proje kapanış koşulları sağlandığında proje tamamlanmış sayılır ve müşteriden signoff otomatik olarak alınmış kabul edilebilir.

Proje sonrasında öğrenilen dersler: Yeni bir projeye başlandığında edinilen tecrübenin kullanılması sağlanır. Uygulama sonrası bir gözden geçirme yapılarak bir kütüphane oluşturulabilir.

Yönetimsel kapanış: Proje kaynaklarının başka bir projeye veya havuza aktarımı yapılır. Ödeme ile ilgili herhangi bir açık konu kalmamalıdır.



PMI (Project Management Institute) ve **PMBOK** (Project Management Body of Knowledge) organizasyonları var.

SOW Statement of Works: Projenin müşterileriyle işi gerçekleştirecek proje ekibi arasında çok genel bir anlaşma metnidir. Genel ürün tanımı, proje kısıtları (zaman, karmaşıklık düzeyi ve teknoloji), iş gereksinimleri, proje katılımcıları rol tanımları, bütçe kısıtları gibi bilgiler içerir.

Maliyet Tahmini: Bilirkişi takdiri, uzman görüşü en çok başvurulan yöntem. En kıdemli kişinin tecrübelerinden faydalanarak verdiği kararlar.

Delphi Yöntemi: Birden fazla uzman tahmin yapar, tahminler birbirine yaklaşıp kadar devam eder.

Analiz Yöntemi: yapılacak iş alt bölümlere ayrılır, ayrı bir yöntem gibi yazdık ama bütün maliyet tahmini metotlarında uygulanırsa daha iyi sonuç alınabilir. Ürün (yazılım modülünün her biri için) ve üretime (analiz, tasarım, gerçekleştirme vs,) bağlı yapılabilir.

Analitik İstatistiksel Yöntem: Yeni yöntemlerdir, yapay zekâ yöntemleri kullanılabilir. CoCoMo (Constructive Cost Model) modeli en çok adı geçen yöntemdir.

CoCoMo – Boehm (Maliyet Tahmin Yöntemi) <-> KLOC ölçütü (1000 kod satır –Line of Code- sayısı, büyüklük ölçütü)

Albrecht-Goffney Modeli<-> FP (Function Point, büyüklük derecesi)

Kemerer Modeli<-> FP (Function Point, büyüklük derecesi)

Büyüklük ölçütlerini doğrudan (KLOC) ve dolaylı (FP) yoldan ölçülenler olarak 2'ye ayırıyoruz.

Kodun kalitesi için de bir kriter olabiliyor. KLOC başına hata gibi.

$$FP = \text{Sayı toplamı} (0.65 + 0.01 \sum Fi) (i = 1 \dots 14)$$

Bütün maliyet tahmini modellerinde iş gücü (E) tahminlemesi yapılıyor.

$$E = A + B \cdot (er)^c$$

er: LOC veya FP

CoCoMo 3e ayrılıyor; temel, orta ve ileri.